TNO report
FEL-97-A023

High level design of the transient detection demonstrator: initial design

TNO Physics and Electronics Laboratory

Oude Waalsdorperweg 63
PO Box 96864
2509 JG The Hague
The Netherlands

Phone +31 70 374 00 00
Fax +31 70 328 09 61

Date
March 1997

Author(s)
M.R. van Velzen

Classification		
Classified by	:	LTZ2OC E.F. Wilms
Classification date	:	5 March 1997
Title	:	Ongerubriceerd
Managementuittreksel	:	Ongerubriceerd
Abstract	:	Ongerubriceerd
Report text	:	Ongerubriceerd

Copy no	:	9	
No of copies	:	21	
No of pages	:	27	(excl RDP & distribution list)
No of appendices	:	-	

All information which is classified according to Dutch regulations shall be treated by the recipient in the same way as classified information of corresponding value in his own country. No part of this information will be disclosed to any party.

The classification designation Ongerubriceerd is equivalent to Unclassified, Stg. Confidentieel is equivalent to Confidential and Stg. Geheim is equivalent to Secret.

The TNO Physics and Electronics Laboratory is part of TNO Defence Research which further consists of:

TNO Prins Maurits Laboratory
TNO Human Factors Research Institute

Netherlands Organization for Applied Scientific Research (TNO)

Managementuittreksel

Titel	:	High level design of the transient detection demonstrator: initial design
Auteur(s)	:	Drs. M.R. van Velzen
Datum	:	maart 1997
Opdrachtnr.	:	A95KM634
IWP-nr.	:	768.1
Rapportnr.	:	FEL-97-A023

In passieve sonar zijn transients per definitie signalen die zelden optreden en (vaak) kort duren. Daar onderzeeboten steeds stiller worden wat betreft het niveau van het stationaire geluid, wordt detectie d.m.v. transients signalen steeds belangrijker bij passieve detectie van onderzeeboten. Voorbeelden van onderzeeboot transients zijn het kraken van de drukvaste huid van een onderzeeboot als deze bezig is met duiken en het geluid dat ontstaat bij het openen van het torpedoluik. Deze transients moeten gedetecteerd worden tegen een achtergrond van zeegeruis en andere transients, meestal afkomstig van zeedieren.
Transients signalen worden voornamelijk met het oor gedetecteerd. Omdat ze kort en zelden aanwezig zijn, worden ze vaak niet opgemerkt door de sonar operator. Daarom is de Koninklijke Marine geïnteresseerd in een automaat die de sonar operator ondersteunt bij de detectie van transients.
De transient detectie demonstrator is een simulatie van een passieve sonar waarin transients optreden. De transients die gewoonlijk worden gedetecteerd door de sonar operator moeten nu worden gedetecteerd door een automaat. De automatische detector presenteert vervolgens het stuk signaal, dat de transient bevat, aan de operator. De operator classificeert vervolgens de aard en herkomst van de transient. Het is de bedoeling dat in de toekomst ook dit classificatieprobleem voor een deel wordt overgenomen door de demonstrator.
De transient detectie demonstrator zal gebruikt gaan worden om verschillende detectie- en classificatiealgoritmen te implementeren [2]. Het beste algoritme zal de hoogste detectiekans hebben bij een gegeven loosalarmkans.
In dit rapport wordt de functionaliteit van de transient detectie demonstrator beschreven met behulp van de Yourdon ontwerpmethode [1]. Deze high level ontwerpmethode wordt gebruikt om de functionaliteit van de software te beschrijven voordat tot de eigenlijke software implementatie wordt overgegaan.
Zo'n ontwerp kan worden gebruikt om met de klant, in dit geval de Koninklijke Marine, het idee en de functionaliteit van het ontwerp te bespreken. In dit stadium is het dan nog eenvoudig om eventuele voorgestelde veranderingen te implementeren.
Tezamen met de toegevoegde introductie, kan het ontwerp ook worden gebruikt om andere wetenschappers snel inzicht te geven in de werking van de demonstrator.

Verder wordt beschreven hoe het Yourdon ontwerp wordt gebruikt om de low level software implementatie te verdelen tussen meerdere ontwerpers.
In het rapport worden eerst de problemen besproken welke optreden bij het detecteren van transients d.m.v. passieve sonar en de algemene visie gegeven op de transient detectie demonstrator.
Hierna worden, met een algemene beschrijving, de Yourdon diagrammen en de definities van de belangrijkste variabelen gepresenteerd. Met behulp van een tekenpakket is ook de uiteindelijke scherm output weergegeven.
Tenslotte wordt beschreven hoe de low level software implementatie wordt verdeeld tussen twee ontwerpers.

Contents

1. Introduction to the transient detection demonstrator

1.1 Introduction to transient detection

One way of detecting submarines is through the use of passive sonar. Detection is done either through listening to the noise produced by submarines or by Fourier processing the noise and presenting the result on a sonar display. The sonar display is called a lofargram and is in fact a time/frequency display. Detection concentrates on, so called, tonal signals emitted by submarines. A tonal is a fixed frequency line and originates from some device within a submarine or surface vessel with periodic behaviour.
The sound and especially the tonal signals produced by submarines has been steadily reduced through the use of modern technics. This is the reason why the Royal Netherlands Navy is searching for other technics for detecting submarines. One can use active sonar to detect submarines but it is clear that this has also specific disadvantages, the most important one being that one betrays oneself. Especially for submarines, active detection is not an attractive option for detecting other submarines or surface vessels.
A submarine produces, besides tonal signals, also transient signals. Transient signals produced by submarines are, for example, the sound produced by the pressure hull of the submarine when the submarine is diving or the sound produced by a torpedo when it is being launched. Transient signals are generally badly suppressed and can therefore be detected at large distances with a good signal-to-noise ratio. Besides submarine transients there are other transient signals present in the ocean. Examples are the noises produced by sea life, like the cry of a whale or a dolphin, or by the surroundings, like the sound made by cracking ice.
A difficulty in submarine transient detection originates from the fact that a particular transient occurs generally only once or twice, as opposed to tonals which are stationary. This increases the probability that a transient is missed by the sonar operator. A further problem is that the transient is often short in duration and/or broad band. Due to the long integration times used in standard passive sonars, the energy of the transient is smeared out, leading to a low signal-to-noise ratio on the lofargram. A transient is often detected and classified by the ear. If we assume that the passive sonar system is capable of beamforming then still the sonar operator can only listen to one beam direction at a time. This also increases the probability that the transient is missed by the sonar operator. The transient project aims to build an automaton that should aid the operator in detecting transient signals. The simplest way of detecting transients is by means of a threshold detector which triggers when some signal excess occurs. Often a transient is actually detected through classification rather than through the strength of the signal. The transient is then detected by specific features that contrast the transient from the background. For this reason classification will eventually become part of the detection process. One should keep in mind, however, that the human operator is superior in classifying a transient, often using knowledge gained from training or experience.

Classification by means of a computer is a field of research that is rapidly evolving. The performance does, however, not match that of humans and it is not very likely that it will in the near future. For this reason, in the current implementation of the detector, the operator will still be the one who makes the final judgement about the origin of the transient. Note, however, that an automaton is superior to the human operator in one respect: it can look for transients in all bearing directions at the same time.

1.2 General view on the transient detection demonstrator

The above remarks have lead to the following considerations concerning the automatic transient detector: The detector should be able to detect a transient signal from the background noise. The background noise can consist of standard ambient noise having a Gaussian character or it could consist of, for example, a more or less continuous background of shrimp noise. When the detector has detected a transient in some bearing direction, it should select a piece of signal centred around the transient and present that to the operator. The operator should then decide whether the signal consists of background noise or contains a transient. When the detector has falsely returned some piece of noise instead of a transient signal, this will be called a false background alarm.
Suppose that the detector is perfectly able to discriminate between the background and the transient signals. Then the operator will be warned if a submarine transient is detected but also if some sea life transient has occurred. Of course the operator is not interested in the sea life transients and we will refer to this as a false transient alarm. The aim of the automatic classification as part of the detector is to suppress these false transient alarms.
Summarising the above: The detector must distinguish transient signals from the background noise, the classifier must distinguish unknown transients from background transients and the human operator finally classifies and interprets the unknown transient.
In a practical implementation the detector should first return all types of transients. The operator should then tell the detector to which class the transient belongs: to the known background transients or to an unknown class which the operator wants to hear each time it occurs. If the transient belongs to the known background transients then the detector should, in some way, suppress the transient when it occurs the next time. In this manner a database is build on-line. It is clear that the more transients are suppressed the higher the probability that a submarine transient is wrongly classified as a background transient. The trade off should therefore be between the number of times a sonar operator is bothered by the false transient alarms and the probability of falsely classifying a transient as a background transient. This is the reason why, when the number of false transient alarms is brought back to an acceptable level, the sonar operator should no longer label transients as background transient.

Although there are basically only two classes of transient signals: known background transients and unknown transients, the known transients should never the less be subdivided in more classes by the operator. The reason for this is that the operator may, after some time, get the impression that the number of transient warnings gets too little. This may occur, for example, when a ship sails from an area with a lot of sea life to an area with much less sea life. The operator may then come to the conclusion that the detector has been tuned to sharp for that area and decide to no longer suppress certain classes of sea life transients, thereby increasing the probability of correctly classifying real transients.
The above construction of the detector treats detection and classification as separated topics. We know that classification can sometimes be part of the detection process. But even if some sort of classification is part of the detection process, it is conceivable that the detection is based on more general rules than the final classification. The point that should be clear is that in the current implementation of the detector, classification is either part of the detection process or it is used as a means of false transient alarm reduction. It is not meant to classify the final submarine or surface vessel transients. These important and rarely occurring transients are left to the human operator.

2. High Level Design

2.1 Description of the high level design

Here we describe the overall working of the transient detection demonstrator and refer to the Yourdon diagrams for the names of the databases and routines denoted within (“ ”). All initial data creation, processing and display parameters are contained in one parameter file (“parameter_file”).

First a signal is constructed consisting of only noise (“data_generation”). Then transients are superimposed onto this signal starting at prespecified time moments. The transients will be scaled to an appropriate signal-to-noise ratio (after beamforming). By scaling the transient and not the noise level, the noise level will be constant within regions where there is no transient signal. The noise can consist of Gaussian noise or it can originate from some file containing background transient noise, consisting of, for example, shrimp noise. When the noise is collected from a file (“transient_database”) we will have to patch the data a number of times to cover the whole range of the signal (“length_of_noise”). If beamforming is desired, each transient is delayed according to some prespecified direction, in such a way that beamforming enhances the signal only in that direction. Because transients are in general broad band signals, a frequency dependent delay will be necessary. The data is written to file (“hydrophone_signals”).
The hydrophone signals are read from file and beamformed (“beamforming”). The beamformed data is further processed and written to file (“return_data_store”). The data written to file will be used to create audio data.
Before being processed the signal is first pre-whitened (“pre_whitening”). The reason for this is that the superimposed transient signal usually also contains some noise and therefore at these places the noise level can suddenly increase. The prewhitened data is also written to file (“return_data_store”) and will be used to present raw unprocessed data to the operator.
The processing (“processing”) of the pre whitened data may consist of, for example, a short time FFT transformation or a wavelet decomposition. The processed data is written to file (“return_data_store”) and is used to present the processed data to the operator.
The processed data is input to a detector (“detection”) which in some manner searches for a transient signal. It should be possible to implement different algorithms within the demonstrator. The optimal algorithm will be a trade off between the number of detections and the number of times the operator will be bothered for false reasons. In the present version of the demonstrator, both a background transient and the submarine transient are presented to the operator as detections. In a future extension of the demonstrator the operator should only be warned if the transient does not match with the known background transients. We will then thus have to include classification into the transient detection demonstrator. The optimal detection/classification algorithm will still be a trade off between the number

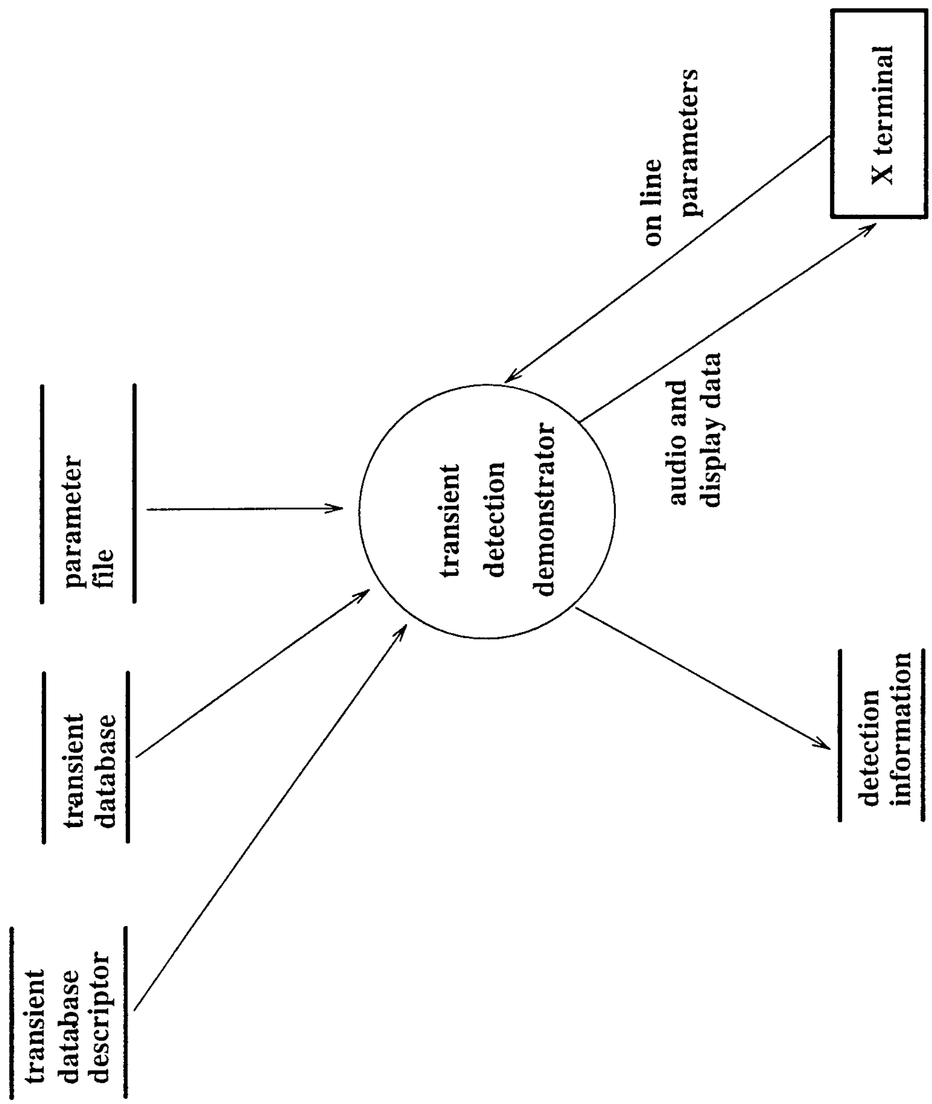
of detections and the number of times the operator will be bothered for false reasons, now including sea life detections as false alarms.
When a detection has occurred, the detector writes general information concerning the detection into a file ("detection_information") and writes also the time of detection into a file ("detection_times"). This last file is read by a display program ("format_audio_and_display_data") which collects patches of data from the database ("return_data_store") around the time of detection and formats the data suitable for display. We will, however, also implement a mode for analysis purposes, in which data is returned continuously within some bearing direction.
The returned patches of data are presented on an X-terminal ("X_terminal") in various plots and the corresponding sound is played on some audio device. The operator can interact ("on_line_parameters") with the display program via a command window ("see X_terminal").
The sound of the transient signal is also a way of signalling to the operator that a detection has occurred.

The program "data_generation" will be written in C.
The routines "beamforming", "pre_whitening", "processing" and "detection" will all be written as a single program in Matlab.
The formatting routine "format_audio_and_display_data" together with the whole X-windows interface will be written as a single program in Matlab.

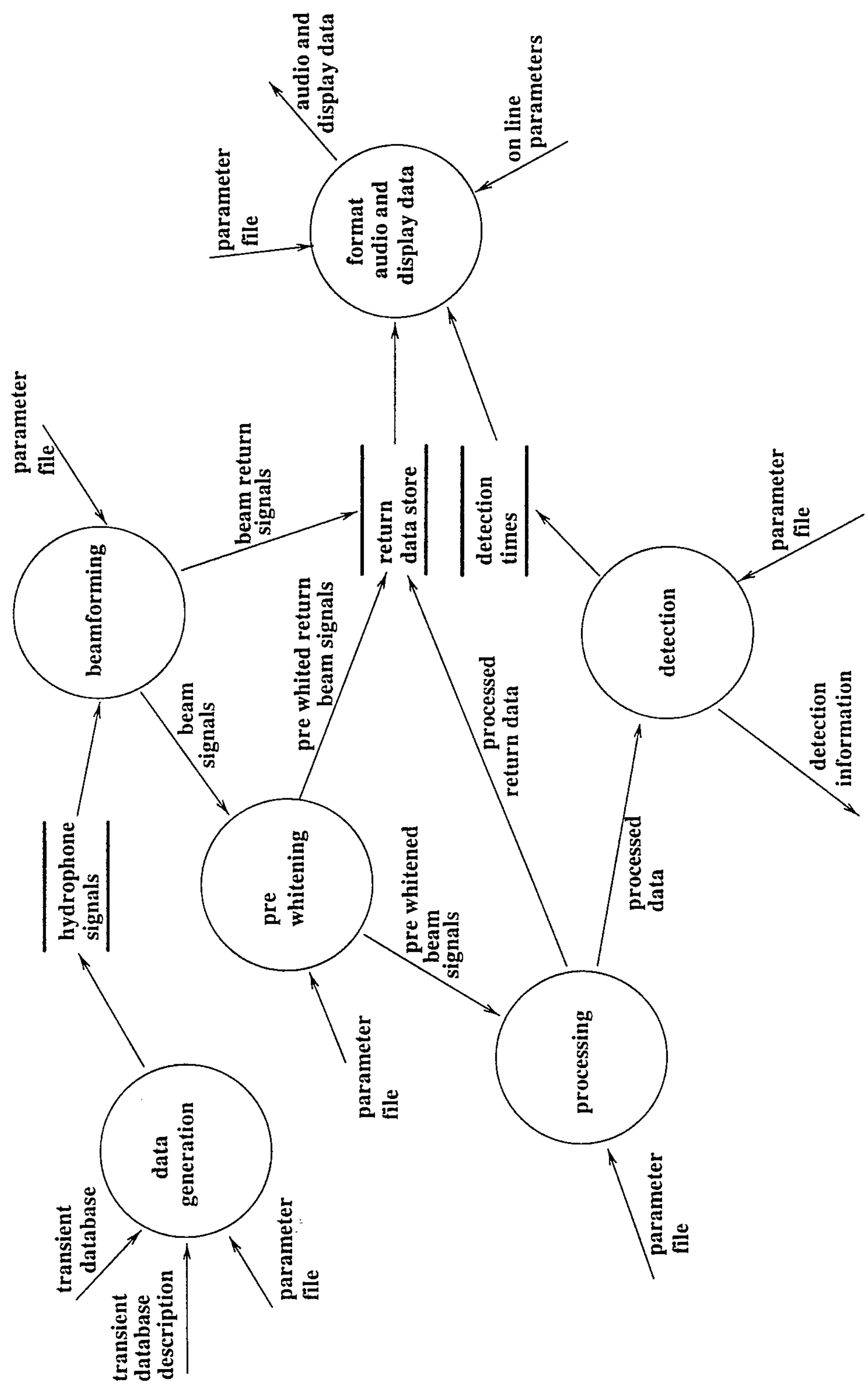
All three programs must be able to run separately on the same machine or on different machines with only the input and output data shared between the machines.

2.2 Diagrams

2.2.1 Context diagram



2.2.2 Transient detection demonstrator



transient_detection_demonstrator (parent: context diagram)

2.3 Variables and Routines

2.3.1 audio and display data

For the following value of the parameters:

display_low_time=100 (s)
display_high_time=120 (s)

display_low_amp=-30 (dB)
display_high_amp=90 (dB)

display_low_freq_amp=30 (dB)
display_high_freq_anp=50 (dB)

display_low_freq=300 (Hz)
display_high_freq=500 (Hz)

display_low_angle=20 (Deg)
display_high_angle=40 (Deg)

display_freq=410 (Hz)

display_angle=35 (Deg)

audio_low_volume=25 (dB)
audio_high_volume=95 (dB)

The displays and the command window are plotted under the definition of the variable "X_terminal".

The variable "display_angle" and "display_freq" are given by the processor if "display_mode = operator_mode" and set by the user if "display_mode = analysis_mode".

2.3.2 beam return signals

The total length of one block of data equals:

(length of header)+angle_data_length*4*number_of_angles
with (length of header)=5*8=40 bytes

The format of one block of data is as described below:

THE HEADER

4 bytes integer: block number

4 bytes integer: year
4 bytes integer: Julian day
4 bytes integer: time of day, the seconds
4 bytes integer: time of day, the microseconds
Thus the time of first sample of the
data block is: seconds+microseconds/1e6

THE DATA

angle_data_length number of 4 bytes IEEE floats (angle: 1)
angle_data_length number of 4 bytes IEEE floats (angle: 2)

.

.

.

angle_data_length number of 4 bytes IEEE floats (angle: number_of_angles)

2.3.3 beamforming

Beamforms the hydrophone signals using broadside beamforming. This means that each frequency component in the signal is delayed in the appropriate manner. Hydrophone array parameters and angle parameters are taken from the "parameter_file". After beamforming the beamformed data is written to the "return_data_store". This is the data that will be presented, after some formatting, to the operator as audio sound.

2.3.4 beam signals

The beamformed data contained in some allocated memory within the overall Matlab program that consists of "beamforming", "pre_whitening", "processing" and "detection".

2.3.5 data generation

Creates noise consisting of Gaussian noise or real noise coming from a file, like shrimp noise. When the noise is collected from a file ("transient_database") we will have to patch the data a number of times to cover the whole range of the signal ("length_of_noise").
Places, on prespecified time moments as described in the "parameter_file", transients into this noise data. The noise data is taken from the "transient_database". which is described in the "transient_database_descriptor".
The transients will be scaled to an appropriate signal-to-noise ratio (after beamforming). By scaling the transient and not the noise level, the noise level will be constant within regions where there is no transient signal.
If beamforming is desired ("do_beamforming=1"), each transient is delayed according to some prespecified direction, in such a way that beamforming enhances the signal only in that direction. Because transients are in general broad band signals, a frequency dependent delay will be necessary.
The program outputs blocks of data as described in "hydrophone_signals".

2.3.6 detection

This program searches in some way for transient signals in the processed data. The simplest example way of detecting transient signals is by selecting data that exceed a constant threshold. However, more advanced ways of detection are certainly necessary to discriminate transients from the background. The type of detection processing and other parameters are taken from the "parameter_file". The result will always be the detection of a transient at a specific moment in time. This time instant is written to the "detection_times" file and signals that a detection has occurred. For each detection specific information concerning the detection, time, angle of detection, etc., is written to the "detection_information" file.

2.3.7 detection information

parameter_file

```
forall detections
  amplitude_of_maximum (dB)
  snr_of_maximum (dB)
  time_of_maximum (hh:mm:ss)
  angle_of_maximum (degree)
  frequency_of_maximum (Hz)
end
```

2.3.8 format audio and display data

This routine formats the data in such a way that it is appropriate for display on an X Terminal. The most important function is to collect data, around the time of detection, when a new detection time has been written to the "detection_times" file. This data is taken from the blocks of data contained in the "return_data_store" files. The amount of data ("return_time_interval") that is returned for one detection is obtained from the "parameter_file".

The routine also resamples the "beam_return_signals" contained in the "return_data_store" in such a way that it is in the appropriate format to be played on some audio device.

The sound of the transient is also a mechanism by which the operator is signalled that something has occurred and that the operator should take a look at the display or replay the sound. Through the "on_line_parameters" the operator is able to interact with this routine. For example: When a new transient is detected, while the previous one is still being investigated by the operator, then the number of detections, which is displayed on the screen, is enlarged. When finished the operator can press the "next" button with the mouse and this routine then starts selecting the new data from the "return_data_store"..

2.3.9 hydrophone signals

Consist of one pair of data files and one pair of communication files: While one data file can be written by "data_generation", the other data file can be read by

"beamforming". Communication about which data file can be read and which can be written is communicated via two files:

If the generic name of data and communication files is called "basename" then the two data files are called "basename_hyd_1.dat" and "basename_hyd_2.dat" while the two communication files are called "basename_com_1.dat" and "basename_com_2.dat".

"basename_hyd_1.com" contains the number 1 if "basename_hyd_1.dat" can be written by "data_generation" and contains the number 2 if it can be read by "beamforming".

"basename_hyd_2.com" contains the number 1 if "basename_hyd_1.dat" can be written by "data_generation" and contains the number 2 if it can be read by "beamforming".

In the initial configurations communication file 1 contains the number 1 and communication file 2 contains the number 1.

"data_generation" writes number 2 while "beamforming" writes numbers 1.

Since the processor is not real-time, both "data_generation" and "beamforming" wait until they get access to the data files.

The total length of one data file equals:

(length of header)+hydrophone_data_length*4*number_of_hydrophones
with (length of header)=5*8=40 bytes

The format of the data (called a block) is as described below:

THE HEADER

4 bytes integer: block number
4 bytes integer: year
4 bytes integer: Julian day
4 bytes integer: time of day, the seconds
4 bytes integer: time of day, the microseconds
Thus the time of first sample of the
data block is: seconds+microseconds/1e6

THE DATA

hydrophone_data_length number of 4 bytes IEEE floats (hydrophone: 1)
hydrophone_data_length number of 4 bytes IEEE floats (hydrophone: 2)

.

.

.

hydrophone_data_length number of 4 bytes IEEE floats (hydrophone: number_of_hydrophones)

2.3.10 on line parameters

This file contains the following parameters:

display_low_time (s)
display_high_time (s)

display_low_amp (dB)
display_high_amp (dB)

display_low_freq_amp (dB)
display_high_freq_anp (dB)

display_low_freq (Hz)
display_high_freq (Hz)

display_low_angle (Deg)
display_high_angle (Deg)

display_freq (Hz)

display_angle (Deg)

audio_low_volume (dB)
audio_high_volume (dB)

See also: audio_and_display_data

2.3.11 parameter file

The file is built as follows:

basename % (String)
% This is the basename of all the data, communication and detection
% information filenames. For example if the basename is called "filename"
% the hydrophone data files are called filename_hyd_1.dat and
% filename_hyd_2.dat

display_mode (operator_mode (1) or analysis_mode (2))

noise_mode (Gaussian (1) or sea life (2))

```
noise_identifier (only read if noise_mode equals 2)
length_of_noise (s)
sample_frequency (Hz)
cut_off_frequency (Hz)
noise_level (dB)
number_of_transients
foreach transient
  transient_number
  start_of_transient (s)
  snr_transient (dB)
  angle_of_transient (degree)
end
do_beamforming (0|1)
number_of_hydrophones
hydrophone_spacing (m)
number_of_angles
proc_low_angle (degree)
proc_high_angle (degree)
angle_data_length (samples)
hydrophone_data_length (samples)
processing_data_length (samples)
blocks_in_store
fs_audio (Hz)
processing_patch_length (s)
processing_algorithm (1: nothing, 2: stft)
processing_parameters (5)
 %For the stft:
 FFT length (samples)
 overlap_factor
return_time_interval (s)
log_of_data (0|1)
do_pre_whitening (0|1)
pre_whitening_algorithm (1: high pass filter)
pre_whitening_parameters (5)
 % High pass filter
 high_pass_cut_frequency (Hz)
detection_parameters
 % Initially:
 detection_threshold (dB)

init_year (1996)
init_julian_day (1-365)
init_time (hh:mm:ss)

%All parameters of the file on_line_parameters
```

%with init_ before the variables

init_display_low_time (s)
init_display_high_time (s)

init_display_low_amp (dB)
init_display_high_amp (dB)

init_display_low_freq_amp (dB)
init_display_high_freq_anp (dB)

init_display_low_freq (Hz)
init_display_high_freq (Hz)

init_display_low_angle (Deg)
init_display_high_angle (Deg)

init_display_freq (Hz)

init_display_angle (Deg)

init_audio_low_volume (dB)
init_audio_high_volume (dB)

2.3.12 pre whitened beam signals

The pre whitened and beamformed data contained in some allocated memory within the overall Matlab program that consist of "beamforming", "pre_whitening", "processing" and "detection".

2.3.13 pre whitened return signals

Same format as the original "beam_return_signals".

2.3.14 pre whitening

Due to the fact that the transients, which are superimposted on the stationary noise, contain some noise, the noise level may increase at these places. Therefore, the data is first pre whitened to obtain stationary noise. A high pass filter, which subtracts the slowly varying trend, is the simplest implementation. The type of processing and the processing parameters are obtained from the "parameter_file". This pre whitened noise is written to the "return_data_store". This is the unprocessed data that will be shown to the operator.

2.3.15 processed data

The processed data contained in some allocated memory within the overall Matlab program that consist of "beamforming", "pre_whitening", "processing" and "detection".

2.3.16 processed return data
Same format as the beam_signals with "processed_data_length" instead of "angle_data_length"

For the STFT the data has the following meaning

|FFT1|FFT2|FFT3|...|FFTn| for one angle, etc. for the other angles.

The time of the header is the time of the first sample which contributed in FFT1. If the FFT overlaps, the display should correctly take this into account when displaying the data to the operator.

2.3.17 processing
Processing may consist, for example, of the short time Fourier transform (stft) or it can consist of the wavelet transform (wt). Type of processing and processing parameters are specified in the "parameter_file". Also certain time-frequency distributions are possible candidates. We assume, however, that the output is time frequency data. This time-frequency data is written to the "return_data_store" and will be shown to the user as processed data in various time-frequency plots.

2.3.18 return data store
This database consist of three pairs of data files and one pair of communication files. If the generic name of data and communication files is called filename then:

The data files which can all be read by "format_audio_and_display_data" are called:

filename_beam_1.dat : written by "beamforming"
filename_prebeam_1.dat : written by "pre_whitening"
filename_proc_1.dat : written by "processing"
filename_beam_2.dat : written by "beamforming"
filename_prebeam_2.dat : written by "pre_whitening"
filename_proc_2.dat : written by "processing"

The two communication files are called:

filename_return_1.com
filename_return_2.com

filename_return_1.com contains the number 1 if datafiles 1 can be written and contains the number 2 if it can be read.

filename_return_2.com contains the number 1 if datafiles 2 can be written and contains the number 2 if it can be read.

In the initial configurations communication file 1 contains the number 1 and communication file 2 contains the number 1.

"processing" writes the number 2 while "format_audio_and_display_data" writes the number 1.

Since the processor is not real-time, all "beamforming", "pre_whitening", "processing" and "format_audio_and_display_data" wait until they get access to the data files.

The three files consist of:

One file with a blocks_in_store number of blocks of: processed data
One file with a blocks_in_store number of blocks of: beamformed data
One file with a blocks_in_store number of blocks of: beamformed and pre whitened data

2.3.19 transient database

Files containing transient signals.

The transient data in the files is collected from a GAC tape (iyxbqpdolo19951115143235) The format of the filenames is: gac1_examplenumber_subdivision.wav. It contains the sound of shrimps, wales and some sounds from submarines. A description is given in: iyxbpqdolo19961205153624

The files are (currently) in directory: /fusr/velr5/transients/tran_data on the f44sn1

The format of the data is: Microsoft Windows 3.1 WAV format sound files. The data in the files consist of is 16-bit data. The sampling frequency is: 22050 Hz

On can look and listen to the sound via program: trans_sounds.m which is also in directory: /fusr/velr5/transients/tran_data

The data is read with program: wavread1. This program can be found in directory: /opt/matlab/toolbox/TNOolbox/wavread1.m

For more information: Type in matlab: help wavread and help wavread1. Read program: trans_sounds.m

The essential code of program tran_data is given below:

```
% Calculate length of data but subtract the header (100).
len=str2num(length_of_file_in_bytes)/2-100;
```

```
% Read the data.
y=wavread1(filename,1,len);

% Resample to approximately the 8192 Hz sample
% frequency used by the SUN
z = resample(y,2,5);

% Play the sound.
sound(z)
```

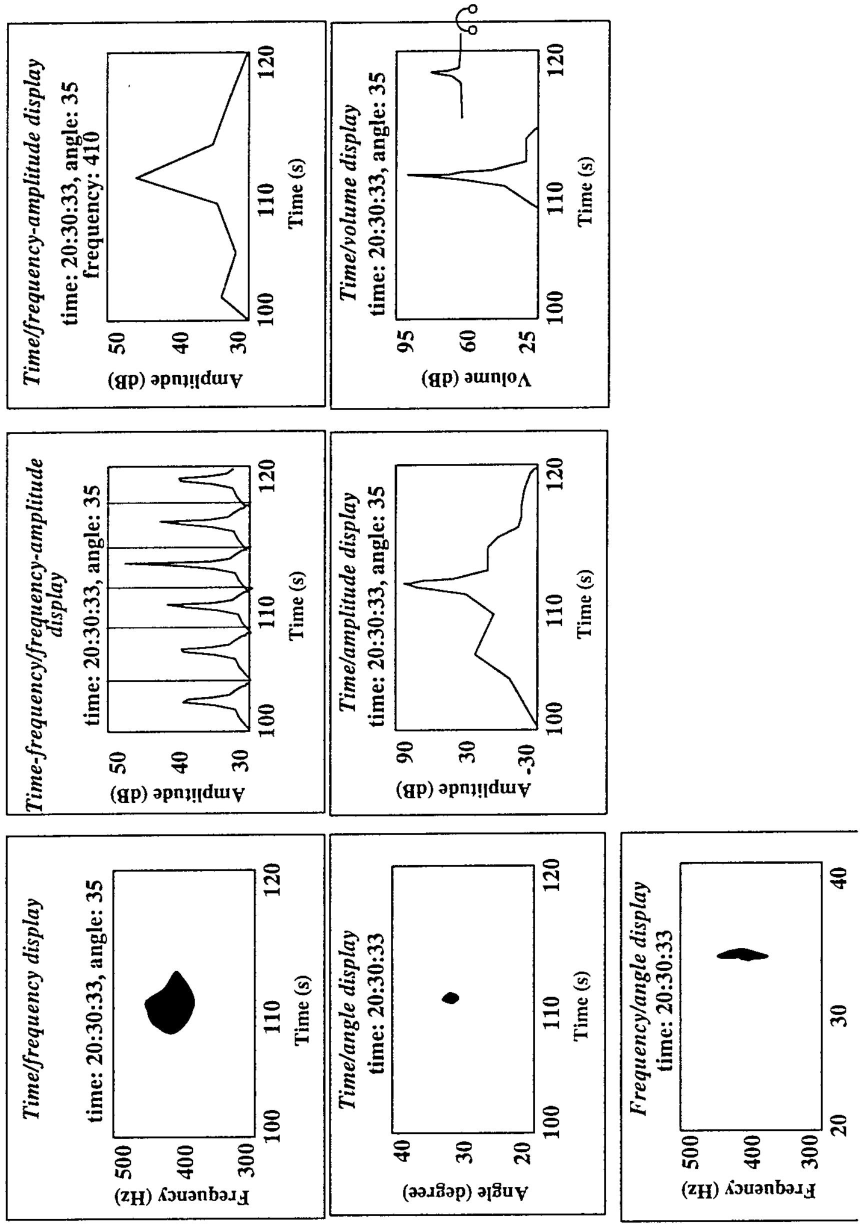
2.3.20 transient database descriptor

The contents of the descriptor file:

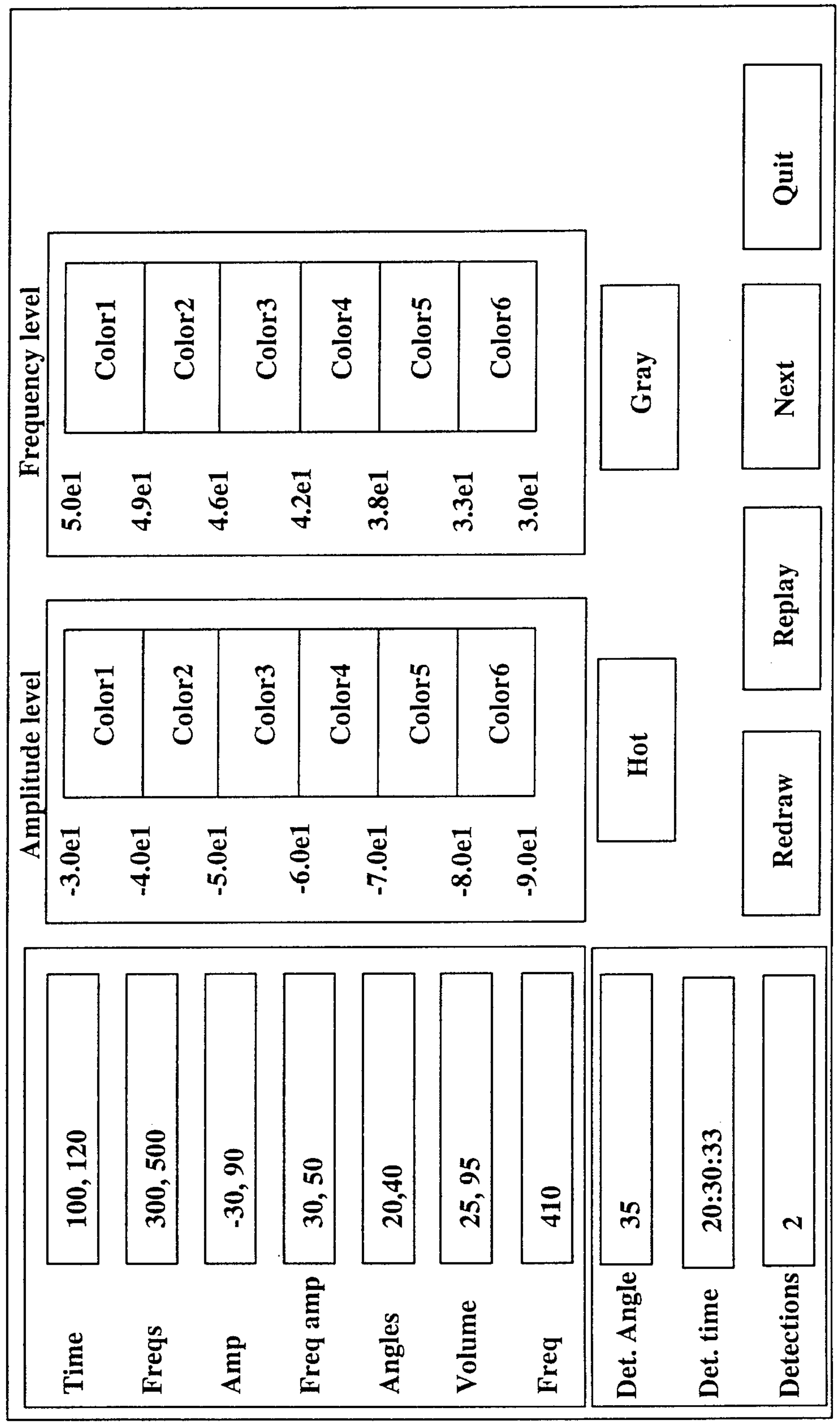
```
# number
filename
sample frequency (Hz)
total length (samples)
% Contents description
% Contents description

# number
filename
sample frequency (Hz)
total length (samples)
% Contents description
% Contents description
```

2.3.21 X terminal

The displays have the following appearance:



The command window has the following appearance:



3. Co-operation on the construction of the transient detection demonstrator

A Yourdon design can be useful to divide the actual implementation of the software between different software engineers. We describe below the parts of Gerard Hotho (GH) and Marcel van Velzen (MvV) in the construction of the transient detection demonstrator.

3.1 Part of GH

First GH creates two data files as described in "hydrophone_signals". The data may consist of Gaussian noise only but may also contain a LFM signal in the data of one hydrophone. GH also creates the two communications files which contain the numbers 1 and 2 (see "hydrophone_signals").
GH writes a faked beamforming program which does the following:

```
beamforming(hydrophone_signals,beam_signals,number_of_hydrophones,
            number_of_angles)
{
        Returns in beam_signals the data of the first "number_of_angles"
        hydrophones, if "number_of_angles" <= "number_of_hydrophones"

        Returns in beam_signals the data of the all hydrophones plus
        ("number_of_angles"-"number_of_hydrophones") times the data of the
        last hydrophone, if "number_of_angles" > "number_of_hydrophones"
}
```

The data in "beam_signals" is now considered as beamformed data. The main program consist of the following parts:

```
main()
{
        Read one block of hydrophone data from one of the data files.

        Update the communication file to allow "data_generation" to write
         to that data file.

        Beamforming (faked: as explained above)

        Write one block of beamformed data to "return_data_store"

        Prewhitening (nothing at the moment).
```

Write one block of prewhitened data to "return_data_store".

The stft of the prewhitened data.

Write one block of processed data to "return_data_store".

Search by means of a constant detection threshold for a detection. If a detection has occurred write the time of detection into the file "detection_times".

if a "blocks_in_store" number of blocks have been written, update communication files to allow "format_audio_and_display_data" to read this data

}

3.2 Part of MvV

MvV writes program "data_generation". This program creates real hydrophone data which is written to file "hydrophone_signals" and which must replace the simple hydrophone signals created by GH.

MvV writes a broad band beamforming program that can be substituted for the fake beamforming program described above.

MvV will display the data in "return_data_store" on an X-terminal with the same output as described under the variable "X_terminal". Special attention must be given to detections that fall within the same "return_time_interval".

4. References

[1] D.J. Hatley and I.A. Pirbhai
Strategies for Real-Time System Specification
1987. Published by Dorset House Publishing Co.,
Inc., 353 West 12th Street, New York, NY 10014
ISBN: 0-932633-11-0

[2] G.H. Hotho
Literature survey on transient detection and classification
To be publised as a TNO-FEL report

5. Signature

The author likes to thank G. Bevers for his technical assistance with software running on PCs.

Dr E.C. van Ballegooijen	M.R. van Velzen
Group leader	Project leader/Author

REPORT DOCUMENTATION PAGE
(MOD-NL)

1. DEFENCE REPORT NO (MOD-NL)	2. RECIPIENT'S ACCESSION NO	3. PERFORMING ORGANIZATION REPORT NO
TD97-0076		FEL-97-A023
4. PROJECT/TASK/WORK UNIT NO	**5. CONTRACT NO**	**6. REPORT DATE**
24826	A95KM634	March 1997
7. NUMBER OF PAGES	**8. NUMBER OF REFERENCES**	**9. TYPE OF REPORT AND DATES COVERED**
27 (excl RDP & distribution list)	2	

10. TITLE AND SUBTITLE

High level design of the transient detection demonstrator: initial design

11. AUTHOR(S)

M.R. van Velzen

12. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

TNO Physics and Electronics Laboratory, PO Box 96864, 2509 JG The Hague, The Netherlands
Oude Waalsdorperweg 63, The Hague, The Netherlands

13. SPONSORING AGENCY NAME(S) AND ADDRESS(ES)

Royal Netherlands Navy
DMKM/WCS/OSAO, Van der Burchlaan 31, 2597 PC The Hague, The Netherlands

14. SUPPLEMENTARY NOTES

The classification designation Ongerubriceerd is equivalent to Unclassified, Stg. Confidentieel is equivalent to Confidential and Stg. Geheim is equivalent to Secret.

15. ABSTRACT (MAXIMUM 200 WORDS (1044 BYTE))

Describes the initial high level Yourdon implementation of the transient detection demonstrator. The transient detection demonstrator simulates a passive sonar receiving sea life and submarine transient signals. The aim of the demonstrator is to aid the sonar operator and to automatically detect these transients within a noisy background. Only when a transient is detected, the demonstrator presents data around the time of detection to the sonar operator.
In the current implementation both a sea life transient and a submarine transient will be presented to the sonar operator. In a future development the demonstrator should only return the submarine transients and therefore the demonstrator will have to include classification as part of the detection process.
The demonstrator will be used to study different detection and classification algorithms.

16. DESCRIPTORS	IDENTIFIERS
Passive sonar	Yourdon
Surges	High level design
Submarine	Demonstrator
	Transients

17a.SECURITY CLASSIFICATION (OF REPORT)	17b.SECURITY CLASSIFICATION (OF PAGE)	17c.SECURITY CLASSIFICATION (OF ABSTRACT)
Ongerubriceerd	Ongerubriceerd	Ongerubriceerd

18. DISTRIBUTION AVAILABILITY STATEMENT	17d.SECURITY CLASSIFICATION (OF TITLES)
Unlimited Distribution	Ongerubriceerd

Distributielijst

1.	Bureau TNO Defensieonderzoek
2.	Directeur Wetenschappelijk Onderzoek en Ontwikkeling*)
3.	HWO-KL*)
4.	HWO-KLu*)
5.	HWO-KM
6.	HWO-CO*)
7 t/m 9.	KMA, Bibliotheek
10.	KIM, t.a.v. Hoofd Vakgroep E, Prof.dr.ir. F.G.J.M. Absil
11.	DMKM/WCS/OSAO, t.a.v. LTZ1 Drs. C.M. Ort
12.	DMKM/WCS/OSAO, t.a.v. LTZ2OC E.F. Wilms
13.	Directie TNO-FEL, t.a.v. Dr. J.W. Maas
14.	Directie TNO-FEL, t.a.v. Ir. J.A. Vogel, daarna reserve
15.	Archief TNO-FEL, in bruikleen aan M&P*)
16.	Archief TNO-FEL, in bruikleen aan Ir. C. Eberwijn
17.	Archief TNO-FEL, in bruikleen aan Dr. ir. E.C. van Ballegooijen
18.	Archief TNO-FEL, in bruikleen aan Drs. M.R. van Velzen
19.	Archief TNO-FEL, in bruikleen aan Ir. ir. G.H. Hotho
20.	Documentatie TNO-FEL
21.	Reserve
	TNO-PML, Bibliotheek**)
	TNO-TM, Bibliotheek**)
	TNO-FEL, Bibliotheek**)

Indien binnen de krijgsmacht extra exemplaren van dit rapport worden gewenst door personen of instanties die niet op de verzendlijst voorkomen, dan dienen deze aangevraagd te worden bij het betreffende Hoofd Wetenschappelijk Onderzoek of, indien het een K-opdracht betreft, bij de Directeur Wetenschappelijk Onderzoek en Ontwikkeling.

*) Beperkt rapport (titelblad, managementuittreksel, RDP en distributielijst).
**) RDP.